Learning the Parameters of Bayesian Networks from Uncertain Data

Segev Wasserkrug[1], Radu Marinescu[2], Sergey Zeltyn[1], Evgeny Shindin[1], Yishai A. Feldman[1]
[1] IBM Research – Haifa
[2] IBM Research Europe
segevw@il.ibm.com, radu.marinescu@ie.ibm.com, sergeyz@il.ibm.com, evgensh@il.ibm.com, yishai@il.ibm.com

Abstract

The creation of Bayesian networks often requires the specification of a large number of parameters, making it highly desirable to be able to learn these parameters from historical data. In many cases, such data has uncertainty associated with it, including cases in which this data comes from unstructured analysis or from sensors. When creating diagnosis networks, for example, unstructured analysis algorithms can be run on the historical text descriptions or images of previous cases so as to extract data for learning Bayesian network parameters, but such derived data has inherent uncertainty associated with it due to the nature of such algorithms. Because of the inability of current Bayesian network parameter learning algorithms to incorporate such uncertainty, common approaches either ignore this uncertainty, thus reducing the resulting accuracy, or completely disregard such data. We present an approach for learning Bayesian network parameters that explicitly incorporates such uncertainty, and which is a natural extension of the Bayesian network formalism. We present a generalization of the Expectation Maximization parameter learning algorithm that enables it to handle any historical data with likelihood-evidence-based uncertainty, as well as an empirical validation demonstrating the improved accuracy and convergence enabled by our approach. We also prove that our extended algorithm maintains the convergence and correctness properties of the original EM algorithm, while explicitly incorporating data uncertainty in the learning process.

Introduction

Bayesian networks (BNs) (Pearl 1988) provide a powerful framework for probabilistic reasoning. In practice, however, the creation of BNs often requires the specification of a large number of parameters, making it highly desirable to be able to learn these parameters from historical data.

In real-world applications, such historical data often has uncertainty associated with it. A prominent example is when such data is available only in unstructured formats. For example, Wasserkrug et al. (2019) report on work on creating a model for electrical equipment diagnosis, in which much of the data available to train the BN was in textual descriptions written by technicians. In order to be able to utilize this data to train the BN, natural language processing (NLP) tools were used to transform the unstructured data into a structured format necessary for learning BN parameters. Of course, such NLP models are not completely accurate, and have both false positives and false negatives. In addition, most such tools provide confidence measures indicating their level of certainty about the outcome. It is of course desirable to use such confidence levels when learning the parameters of the BN.

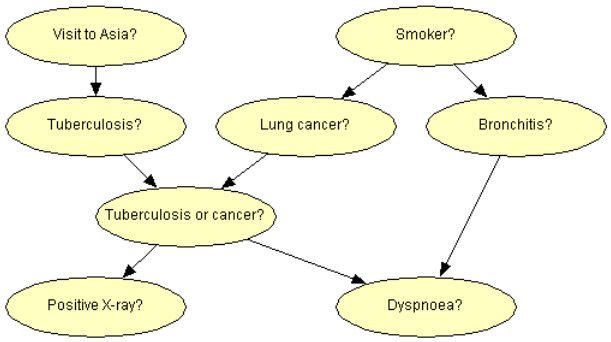
While there are standard ways to incorporate uncertainty during inference in BNs (Mrad et al. 2015), to the best of our knowledge, incorporating uncertainty of any sort during parameter learning has not been addressed.

In this paper, we address this gap. Our primary contribution is an algorithm that enables the use of uncertain data as inputs for learning the parameters of BNs. More specifically, we extend the standard EM algorithm for BNs (Koller and Friedman 2009) to incorporate historical examples with *likelihood evidence*–based uncertainty. In addition, we prove that our proposed algorithm maintains the correctness and convergence properties of the original EM algorithm. Finally, we present an empirical validation that demonstrates the value of our approach.

Preliminaries

A *Bayesian network* (Pearl 1988) is defined as $\mathcal{B} = \langle \mathbf{X}, \mathbf{D}, G, \mathbf{P} \rangle$, where $\mathbf{X} = \{X_1, \ldots, X_n\}$ is a set of random variables, $\mathbf{D} = \{D_1, \ldots, D_n\}$ is the set of the corresponding domains, $G = (\mathbb{V}, \mathbb{E})$ is a directed acyclic graph over $\mathbf{X}$ (that is, $\mathbb{V} = \mathbf{X}$), and $\mathbf{P} = \{P_1, \ldots, P_n\}$, where $P_i = \Pr(X_i \mid \mathrm{Pa}_i)$ is the *conditional probability table* (CPT) that provides the conditional probability of each value of X_i given the values of its parents Pa_i.

Bayesian networks are often used for diagnosis, in both medical (Kahn et al. 1997) and engineering domains (Cai, Huang, and Xie 2017). Figure 1(a) depicts a well-known example of a diagnosis network in the medical domain, which we will use to illustrate the key concepts of our work (and which was also empirically analyzed). The network is often called the "Asia" or "Lung cancer" network (Lauritzen and Spiegelhalter 1988). Figure 1(b) shows the CPT of the node "Dyspnoea?". Once its parameters have been deter-



(a) Asia Bayesian network

Dyspnoea?

Bronchitis?	yes		no	
Tuberculosis or cancer?	yes	no	yes	no
yes	0.9	0.8	0.7	0.1
no	0.1	0.2	0.3	0.9

(b) CPT for the Dyspnoea node

Figure 1: Example of a Bayesian network.

mined, the network can use evidence regarding "Visit to Asia?", "Smoker?", "Positive X-ray?", and "Dyspnoea?" to compute the probability of "Lung cancer?", "Tuberculosis?" and "Bronchitis?".

It is quite conceivable that in real-world medical cases in which such diagnosis BNs need to be created, available historical information is in unstructured format such as written reports or X-ray images. For example, it is quite likely that whether a specific patient has the symptom "Dyspnoea?" or had a "Positive X-ray?" has to be extracted from a description written by a physician. NLP tools can be used to transform such unstructured data into a structured format, with associated uncertainty. For example, an NLP tool run on such a historical report may be able to indicate with a 0.7 confidence level that a specific patient indeed had the "Dyspnoea?" symptom. The work we describe here can effectively utilize such data when learning the parameters of a BN.

Related Work

There are two primary types of relevant previous work. The first consists of various types of algorithms for learning the parameters of BNs, and the second deals with uncertain evidence.

Over the years, a variety of algorithms for learning the parameters of BNs have been proposed. Two prominent types of algorithms are *Maximum Likelihood Estimation* (*MLE*) (Spiegelhalter and Lauritzen 1990) and *Bayesian Learning* (Bernardo and Smith 1994). A primary difference between these is that MLE methods provide point estimates of the parameters, while Bayesian Learning maintains a constantly updated distribution over these parameters (this enables Bayesian Learning to continuously learn and improve the parameters as new examples are provided, as well as incorporate expert estimates).

Expectation Maximization (*EM*) is an MLE-type algorithm that supports learning from evidence with *missing values*, i.e., data in which the values of some of the variables are unknown. An example for the Asia network (Fig. 1(a)) could be an input where values are given for all the nodes in the network except for "Lung Cancer?", which is unknown. EM has played a critical role in learning probabilistic graphical models and BNs (Dempster, Laird, and Rubin 1977; Lauritzen 1995; Heckerman 1998). There have been many enhancements proposed over the years to address a variety of challenging situations for EM, such as slow convergence or the presence of hidden variables (Bauer, Koller, and Singer 1997; Ortiz and Kaelbling 1999; Thiesson, Meek, and Heckerman 2001; Elidan et al. 2002; Elidan and Friedman 2005).

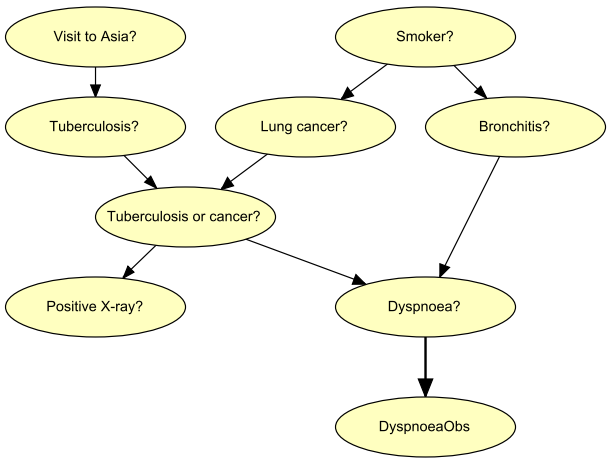
In contrast to the case of missing evidence, which is used when there is missing knowledge about the values of some random variables, uncertain evidence is usually introduced whenever there is knowledge about the value of the random variable, but the observational process is unable to clearly report a single state for the observed variable. Historically, several types of uncertain evidence have been considered in this context (Mrad et al. 2015; Pan, Peng, and Ding 2006). *Likelihood* (or *virtual*) evidence is perhaps the most common kind of uncertain evidence in the context of BNs. Likelihood evidence "corresponds to the cases where the observation is uncertain," where "the uncertainty on the observation may come from the unreliability or imprecision of the source of the information" (Mrad et al. 2015). This type of uncertainty is suitable for representing the uncertainty associated with information sources, such as sensors and unstructured analysis, because likelihood evidence is assumed not to incorporate any prior knowledge beyond what appears directly in the information source (the sensor, text, or picture). This is the type of uncertain evidence that we address in this paper.

Several approaches have been proposed over the years to deal with inference using uncertain evidence in BNs, including Pearl's method, which uses an auxiliary binary child for each variable with likelihood evidence (Pearl 1988), entropy-based techniques (Valtorta, Kim, and Vomlel 2002; Peng, Zhang, and Pan 2010), and recent methods based on credal networks (Marchetti and Antonucci 2018). During such inference, combining multiple sources of evidence and uncertainty is typically done in a Bayesian framework that allows for various prior distributions to be combined in a flexible manner (Spiegelhalter and Best 2003).

However, all of the above work on dealing with uncertain evidence addresses how to incorporate such evidence during BN inference, i.e., after the network's parameters have already been determined. To the best of our knowledge, no previous work has addressed this issue during the learning of the parameters. Song et al. (2012) address a type of uncertainty called *attribute uncertain data*, but only while learning the structure of the BN. No formal proof is provided regarding the properties of this method; the incorporation of uncertainty is based on heuristics, and is carried out in a manner that is external to the BN.

Learning BN Parameters with Uncertainty

In this section, we detail our algorithm for learning the parameters of a BN given a set of cases in which some inputs may



(a) Extended network

DyspnoeaObs

Dyspnoea?	yes	no
false	0.3	0.7
true	0.7	0.3

(b) CPT for the Dyspnoea node

Figure 2: Extending the Asia network to carry out likelihood evidence based inference.

have associated uncertainty of the type *likelihood evidence* (Pearl 1988).

Likelihood evidence is an observation that does not single out a unique value of a variable X that has k possible values, and is represented by a *likelihood ratio* $L(X) = (L(X = x_1) : \ldots : L(X = x_k))$. When this is normalized, $L(X = x_i)$ is $\Pr(\text{obs} \mid X = x_i)$, the probability of the observation occurring when the value of X is x_i. Given this interpretation, BN inference with likelihood evidence can be carried out as follows (Pearl 1988). A virtual node for the observation is added to the network with an appropriate CPT; the virtual evidence is set as a hard finding on this node; the evidence on all the nodes in the BN is then propagated using standard BN propagation algorithms (Pearl 1988; Koller and Friedman 2009).

For example, suppose that an NLP analysis of a historical medical record relevant to the Asia network (Fig. 1(a)) finds the symptom "Dyspnoea?" with 0.7 confidence. This would be represented by assigning the node "Dyspnoea?" in the network the likelihood evidence $(0.7 : 0.3)$. In order to carry out inference with such evidence, the following steps would be taken.

1. The original Asia BN would first be extended with a boolean-valued node "DyspnoeaObs", which would be a child of "Dyspnoea?" (see Fig. 2(a)), and have the CPT shown in Fig. 2(b).
2. The value of "DyspnoeaObs" would be set to true, and inference would be carried out on the augmented network using any standard BN inference algorithm.

We use this method for inference with likelihood evidence to extend the EM algorithm (Koller and Friedman 2009, Sec. 19.2.2) to learn BN parameters from data with likelihood evidence. The original algorithm learns BN parameters with missing data, i.e., on examples for which some of the values of the variables in the network are missing or unknown. To do this, it repeats two steps: an *expectation* step, in which, for each example, the missing data values are replaced with the expected values given the current BN parameters using BN inference; and a *maximization* step, in which the maximum likelihood values of the BN parameters are calculated given the (now complete) data. At a high level, this algorithm can be described as follows:
Repeat until convergence:

1. Complete the data for each example by calculating the expected value for each variable with missing values given the current parameters of the BN.
2. Update the parameters of the BN to the Maximum Likelihood Estimate (MLE) given the set of full data provided by the expectation step.

The core idea of our algorithm is to use inference with likelihood evidence, as described above, in the expectation step. Step 1 above is thus replaced by the following:

1. Complete the data for each example:
   (a) Extend the original network by adding nodes and edges to each node for which there is likelihood evidence, as well as the appropriate CPTs. (In this way, the example with likelihood evidence can be replaced with a new example with only missing data. For example, if an observation child has been added to node V, the node V now has an unknown value in this new augmented data point.)
   (b) Calculate the expected value for the nodes with the missing data.

Algorithm 1 describes our EM-Likelihood approach. It takes as input a Bayesian network $\mathcal{B}$ and a set of data examples S. Each data example $S_j \in S$ contains for each variable X_i one element $d_i \in D_i \cup L_i \cup \{?\}$, where $L_i = \{(\Pr(\text{obs} \mid X_i = x_1) : \ldots : \Pr(\text{obs} \mid X_i = x_k))\}$ is the set of all likelihood evidence values for the k possible values of variable X_i, and "?" denotes the unknown value. That is, the value of variable X_i in any example S_j can be any of its discrete values, unknown, or a new type of value, indicating likelihood evidence.

The purpose of the algorithm is to learn the values for the network's parameters θ; more specifically, for each value x_l of variable X and values $\boldsymbol{u}_l$ of the parents of X in the network, the conditional probability $\theta_{x_l|\boldsymbol{u}_l}$. The set of all $(x_l, \boldsymbol{u}_l)$ values is denoted by $Val(X, \text{Pa}_X^{\mathcal{B}})$. The algorithm computes successive approximations θ^t at each iteration t. Following Koller and Friedman (2009), we use $\bar{M}_{\theta^t}[x_l, \boldsymbol{u}_l]$ to denote the expected sufficient statistics for $x_l \mid \boldsymbol{u}_l$, based on the current set of parameters.

The EM-Likelihood algorithm is based on the EM algorithm appearing in Koller and Friedman (2009), with the primary difference being the expectation step, which takes

Algorithm 1 EM-Likelihood: an EM algorithm for learning with likelihood evidence

Require: Bayesian network $\mathcal{B} = \langle \mathbf{X}, \mathbf{D}, G, \mathbf{P} \rangle$, dataset S

procedure EM($\mathcal{B}$, S)
  $\theta \leftarrow \theta^0$ ▷ Initialize by uniform sampling
  for $t \leftarrow 1, 2, \ldots,$ **until** convergence **do**
    $\bar{M}_{\theta^t} \leftarrow$ COMPUTE-ESS(G, θ^t, S)
    for $i \leftarrow 1, \ldots, n$ **do**
      for each $x_l, \boldsymbol{u}_l \in Val(X_i, \mathrm{Pa}_{X_i}^{\mathcal{B}})$ **do**
        $\theta_{x_l|\boldsymbol{u}_l}^{t+1} \leftarrow \frac{\bar{M}_{\theta^t}[x_l, \boldsymbol{u}_l]}{\bar{M}_{\theta^t}[\boldsymbol{u}_l]}$

function COMPUTE-ESS(G, θ, S)
  for $i \leftarrow 1, \ldots, n$ **do**
    for each $x_l, \boldsymbol{u}_l \in Val(X_i, \mathrm{Pa}_{X_i}^{\mathcal{B}})$ **do**
      $\bar{M}[x_l, \boldsymbol{u}_l] \leftarrow 0$
  for each example $S_j \in S$ **do**
    $(G', \theta') \leftarrow$ AUGMENT-BN(G, θ, S_j)
    $E \leftarrow \emptyset$ ▷ Initialize evidence set
    for $i \leftarrow 1, \ldots, n$ **do**
      if X_i has a unique value $d_i \in D_i$ in S_j **then**
        $E \leftarrow E \cup \{X_i = d_i\}$
      else if X_i has likelihood evidence in S_j **then**
        $E \leftarrow E \cup \{O_{X_i} = \text{true}\}$
    Run inference on (G', θ') with the evidence E
    for $i \leftarrow 1, \ldots, n$ **do**
      for $x_l, \boldsymbol{u}_l \in \mathrm{Val}(X_i, \mathrm{Pa}_{X_i}^{\mathcal{B}})$ **do**
        $\bar{M}[x_l, \boldsymbol{u}_l] \leftarrow \bar{M}[x_l, \boldsymbol{u}_l] + P_{(G', \theta')}(x_l, \boldsymbol{u}_l \mid E)$
  return $\bar{M}$

function AUGMENT-BN(G, θ, S)
  Initialize $G' \leftarrow G$, $\theta' \leftarrow \theta$
  for each variable X with likelihood evidence $(l_1 : \ldots : l_k)$ in S **do**
    ▷ Add new observation node, connect to variable
    $G'_{\mathbb{V}} \leftarrow G'_{\mathbb{V}} \cup o_X$, $G'_{\mathbb{E}} \leftarrow G'_{\mathbb{E}} \cup (X, o_X)$
    $D_{o_X} \leftarrow \{\text{true}, \text{false}\}$
    for $i \leftarrow 1, \ldots, k$ **do**
      $\theta'_{O_X = \text{true} | X = x_i} \leftarrow l_i$
  return (G', θ')

likelihood evidence into account by creating for each example a new BN that has additional observation nodes o_X for each variable X with likelihood evidence, connected as a child of X (function AUGMENT-BN). The CPTs for the new nodes are created based on the likelihood values (function COMPUTE-ESS).

The augmentation of the inference step is seemingly a small change in the original algorithm. However, note that in this algorithm, inference, and, therefore, the resulting parameter learning, is carried out based not on a single network, but rather on a set of networks, each derived from the original network. Moreover, the number of different networks is equal to the number of examples that contain likelihood evidence, and networks differ based on the number and identity of nodes with likelihood evidence. Therefore, this extension requires formal justification, which we provide in the sequel. Specifically, we show that our algorithm converges, and is correct in the sense that it computes a Maximum Likelihood Estimate, i.e., that it converges to a local maximum of $l(\theta : S)$, the likelihood function of the parameters θ of the original BN given the dataset S. This claim is formalized in Theorem 1. First, we define the following notation.

Let $S = \{S_1, \ldots, S_m\}$ be a set of input examples to a BN with a fixed structure $\mathcal{B}$. For each example S_j, let S_j^L be the set of variables with likelihood evidence in S_j, and let S_j^F be the set of variables with fixed values. Let $\mathcal{B}_j$ be the Bayesian network derived from $\mathcal{B}$, augmented with new observation nodes o_X for variables X that have likelihood evidence in the example S_j (as created in function AUGMENT-BN) and the CPTs based on the likelihoods (as computed by COMPUTE-ESS). Define the likelihood function $l(\theta : S) = \log \prod_{j=1}^{m} P_{\mathcal{B}_j}(S_j^F | \theta)$, where $P_{\mathcal{B}_j}(S_j^F | \theta)$ is the probability of getting the deterministic values in example S_j in the Bayesian network $\mathcal{B}_j$, with evidence consisting of all the o_X variables set to true.

THEOREM **1** (correctness and convergence)**.** *The EM-Likelihood algorithm converges to a local maximum of the likelihood function* $l(\theta : S)$.

Proof (sketch). The core idea behind the original correctness and convergence proof of the EM algorithm uses the *expected log-likelihood* function that, given a set of independent partial observations S and a joint distribution Q for creating a complete assignment H to these partial observations, is defined as

$$E_Q[l(\theta : \langle S, H \rangle)] = \sum_H Q(H) l(\theta : \langle S, H \rangle).$$

Then, by the linearity of expectations and the conditional independence relationships on the parameters of CPTs induced by the structure of the BN, it is shown that

$$E_Q[l(\theta : \langle S, H \rangle)] = \sum_{i=1}^{n} \sum_{\substack{(x_i, \boldsymbol{u}_i) \in \\ (X_i, \mathrm{Pa}_{X_i})}} \bar{M}_Q[x_i, \boldsymbol{u}_i] \log \theta_{x_i|\boldsymbol{u}_i},$$

where $\bar{M}_Q[x_i, \boldsymbol{u}_i]$ is the expected count according to Q that the node X_i and its parents Pa_{X_i} have the joint value $(x_i, \boldsymbol{u}_i)$, and $\theta_{x_i, \boldsymbol{u}_i}$ is the CPT entry corresponding to the conditional probability that the value of X_i is x_i given that the values of its parents are $\boldsymbol{u}_i$. It is then shown that selecting the probability Q to be the probability induced by each step in the EM algorithm results in convergence to a local maximum of the the likelihood $l(\theta : S)$.

Similarly, in our case, it can be shown that given a set of observations S' with likelihood evidence, we can define an expected likelihood function $E_Q[l(\theta : \langle S', H \rangle)]$, such that

$$E_Q[l(\theta : \langle S', H \rangle)] = \sum_{i=1}^{n} \sum_{\substack{(x_i, \boldsymbol{u}_i) \in \\ (X_i, \mathrm{Pa}_{X_i})}} \bar{M}_Q[x_i, \boldsymbol{u}_i] \log \theta_{x_i|\boldsymbol{u}_i},$$

where H is the completion of S' with the missing values for the examples of the augmented BNs. This is based on the fact

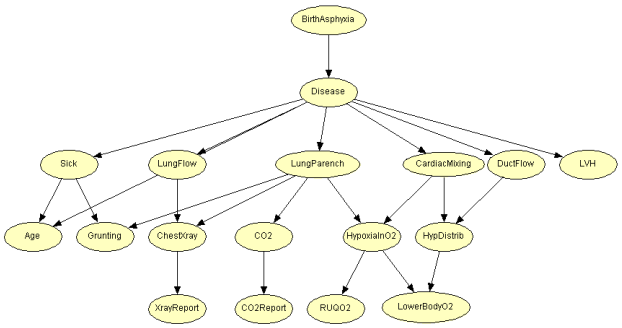


Figure 3: Simplified Child network

that S' can be viewed as a set of missing evidence for a set of related BNs that share a common set of parameters that need to be learnt, and due to the ability to independently maximize the CPT of each node in this set of networks when calculating the overall MLE. Furthermore, we show, analogously to the original EM, that our choice of Q results in a local maximum of the likelihood function $l(\theta : S')$ □

Note that, in addition, it is quite straightforward to augment our algorithm to enable Bayesian Learning benefits such as incorporating expert estimates as well as enabling continuous improvement and updating. This can be done by storing the (expected) number of historical examples used to train the each entry in the CPT, and taking these values into account when updating the parameters when provided with a new set of examples.

Empirical Results

We implemented our algorithm on top of the open-source Merlin[1] library and used three networks for validation. In all experiments we initialized the algorithm using a uniform distribution. We began with a network containing just two binary nodes: "Problem" and its child "Symptom." We then tested it on the Asia network (Fig. 1(a)), and finally on a larger network, which is a slightly simplified version of the Child network (Spiegelhalter and Cowell 1992) (Fig. 3).

For each experiment, we selected nodes with which we associate likelihood evidence. Then we went through the sequence of steps, briefly presented in the following list (we used Hugin[2] to generate samples from the networks).

1. Extend a base network by adding an "observation" child node to each node for which we wanted to generate likelihood evidence.
2. Assign CPTs with the chosen values of likelihood to the "observation" nodes.
3. Sample data for the extended network with the assigned CPTs.
4. Create a dataset with likelihood evidence from the sampled data.

[1]Available at http://github.com/radum2275/merlin.

[2]See http://hugin.com.

	Node	
	Dyspnoea	DyspnoeaObs
Sampled data	true or false	true
Likelihood evidence dataset	0.7	—
Deterministic dataset	true	—

Table 1: Producing likelihood and deterministic datasets from the simulated data.

5. Create a deterministic dataset, against which to compare, from the sampled data.
6. Repeat Steps 2–5 with other likelihood values and the corresponding CPTs.
7. Merge datasets with likelihood evidence that were sampled from different CPTs to a single dataset. Perform the same operation for the deterministic datasets.
8. Run EM parameter estimation for the dataset with likelihood evidence and the deterministic dataset. Compare goodness-of-fit with respect to the actual network CPTs.

To illustrate these steps, consider the "Asia, Dyspnoea" experiment, in which we introduced likelihood evidence only to the node "Dyspnoea?"of the Asia network. In step 1, we added the node "DispnoeaObs" as a child to node "Dyspnoea?" as shown in Fig. 2(a). In Step 2, to generate 0.7 likelihood evidence for "Dyspnoea?" node, the child "DyspnoeaObs" was assigned the CPT from Fig. 2(b). In Step 3, we sampled multiple times from this network to get many examples with likelihoods both of 0.7 and 0.3. A 0.7 likelihood is obtained when the value of the node "DyspnoeaObs" is true and 0.3 when it is false.

In Step 4, from the sampled dataset, we created a dataset with likelihood evidence by placing the relevant likelihoods in the parent of the appropriate observation node in the original network and deleting data for the observation node. To derive the deterministic dataset against which to compare in Step 5, we placed the value of the observation node as the value for the parent node. Table 1 illustrates this process. Note that the sampled value of the "Dyspnoea" node in the sampled data is used neither in the deterministic nor in the likelihood evidence dataset.

In Step 6, we repeated sampling using various settings for the CPTs of child nodes to specify various likelihoods. For each uncertain node, we varied the CPTs (and thereby generated likelihood evidence), over the values 0.6, 0.7, 0.8, 0.9 and 0.95. Then, in Step 7, these observations were merged into a dataset with 100,000 observations per experiment, 20,000 observations per each value for each uncertain node.

In the two-node network, we performed a single experiment with uncertainty on the "Symptom" node. For the Asia network, we ran five experiments. Four experiments simulated networks with one uncertain node per experiment, one each of "Positive X-Ray?", "Dyspnoea?", "Asia?", and "Smoker?". The final experiment simulated a network with likelihood values generated simultaneously for all these nodes. Finally, two experiments were performed with the

Experiment	Overall discrepancy, %			
	Maximum		Average	
	Deter-ministic	Likeli-hood	Deter-ministic	Likeli-hood
Two-Node	17.28	1.00	8.70	0.45
Asia, X-Ray	21.04	0.71	2.61	0.15
Asia, Dyspnoea	17.53	0.85	1.76	0.11
Asia, Asia?	20.62	1.92	2.92	0.22
Asia, Smoker	6.49	0.65	1.10	0.10
Asia, 4 nodes	20.58	1.10	7.95	0.17
Child, Case 1	16.95	2.83	1.12	0.30
Child, Case 2	25.08	2.96	3.99	0.34

Table 2: Overall discrepancies in EM likelihood experiments.

Node	Discrepancy, %			
	Maximum		Average	
	Deter-ministic	Likeli-hood	Deter-ministic	Likeli-hood
Problem	0.36	0.36	0.36	0.36
Symptom	17.28	1.00	17.04	0.54

Table 3: CPT discrepancies for the two-node network.

Child network. In the first one (Case 1), uncertainty was assigned to the "Birth Asphyxia" node, which is not a leaf node. In the second experiment (Case 2), uncertainty was assigned to four nodes: "Birth Asphyxia", "LVH", "Grunting", and "HypDistrib".

In order to measure the goodness-of-fit in all experiments, we computed for each CPT the average and maximum absolute differences between the probability estimates and their actual values for each set of values of a CPT of a node and its parents. We then looked at the average of these averages and maximum of these maxima. These are named "average overall discrepancy" and "maximum overall discrepancy," respectively.

Table 2 summarizes the overall discrepancies of the experiments. In all experiments, parameter estimates that were obtained via our EM-Likelihood algorithm are close to the actual network parameters. In contrast, approximation using deterministic values results in a significant deviation from the original parameters.

Table 3 provides CPT discrepancies for nodes of the two-node network. The deterministic dataset generates strongly biased estimates for the "Symptom" node, whereas discrepancies for the likelihood dataset are small. In contrast, the "Problem" node is not affected by uncertainty, and the discrepancies for this node are small and identical for both datasets. This is to be expected, since the same deterministic sampled values for this node and its parents are used for creation of both likelihood and deterministic datasets. In general, as in our sampling methodology nodes either have actual values or likelihood evidence assigned, nodes that are not assigned likelihood evidence and are not descendants of a node assigned likelihood evidence, are not affected by such evidence because intermediate nodes have actual values.

Node	Discrepancy, %			
	Maximum		Average	
	Deter-ministic	Likeli-hood	Deter-ministic	Likeli-hood
Smoker	0.00	0.02	0.00	0.02
Tuberculosis	0.30	0.30	0.15	0.15
Lung cancer	1.90	0.09	1.89	0.07
Bronchitis	6.49	0.29	6.40	0.20

Table 4: CPT discrepancies for the Asia network. Uncertain data for the "Smoker" node.

Node	Discrepancy, %			
	Maximum		Average	
	Deter-ministic	Likeli-hood	Deter-ministic	Likeli-hood
Visit to Asia	17.42	0.04	17.42	0.04
Tuberculosis	3.87	0.96	1.97	0.50

Table 5: CPT discrepancies for the Asia network. Uncertain data for the "Visit to Asia" node.

Table 4 shows CPT discrepancies for several nodes of the Asia network in the experiments with uncertainties for the "Smoker" node. Note that the discrepancies for "Smoker" itself are minor. Since the CPT of "Smoker" is $\binom{0.5}{0.5}$, its marginal distribution does not change following uncertain data sampling. However, CPT estimates of its children, "Lung cancer" and "Bronchitis", have large discrepancies for the deterministic dataset. "Tuberculosis" is not a child of "Smoker", so its discrepancies are small for both approaches, as expected.

Table 5 summarizes experiments with uncertainty for "Visit to Asia". For the deterministic dataset we observe significant discrepancies for "Visit to Asia" and its child, "Tuberculosis".

Table 6 shows CPT discrepancies for all nodes of Asia network in the experiments with 4 uncertain nodes. In this case, all nodes, except for the logical node "Tuberculosis or cancer", are either uncertain, or one of their children is. We observe very significant differences in the discrepancy measures for the deterministic and likelihood datasets.

Table 7 presents the detailed results of two experiments with the Child network. We show discrepancies for the nodes with assigned uncertainty and their children. "Disease" is a child of "Birth Asphyxia", and "Lower Body O2" is a child of "Hypoxia Distribution". We again observe that the likelihood evidence EM algorithm provides good results.

We also ran a convergence experiment on the two-node network by varying the number of samples between 1,000 and 100,000. The results displayed in Fig. 4 indicate that the goodness-of-fit for the likelihood evidence datasets gradually improves with the sample size and is satisfactory even for small sample sizes. In the deterministic setting, not only do the estimates deviate significantly from the actual network parameters, but there is no significant improvement even for large sample sizes.

Node	Discrepancy, %			
	Maximum		Average	
	Deterministic	Likelihood	Deterministic	Likelihood
X-Ray	20.11	0.43	19.61	0.36
Dyspnoea	16.97	1.10	13.34	0.56
Visit to Asia	20.58	0.06	20.58	0.06
Smoker	0.04	0.02	0.04	0.02
Tuberculosis	3.87	0.25	1.95	0.14
Lung cancer	1.94	0.13	1.85	0.09
Bronchitis	6.21	0.13	6.20	0.11
Tuberculosis or cancer	0.00	0.00	0.00	0.00

Table 6: CPT discrepancies for the Asia network. Four nodes with uncertain data.

Node	Discrepancy, %			
	Maximum		Average	
	Deterministic	Likelihood	Deterministic	Likelihood
Birth Asphyxia, Case 1	16.95	0.09	16.95	0.09
Birth Asphyxia, Case 2	16.83	0.13	16.83	0.13
Disease, Case 1	12.28	2.20	2.19	0.46
Disease, Case 2	12.17	1.02	2.14	0.33
LVH, Case 2	18.86	0.78	17.21	0.26
Grunting, Case 2	18.90	1.35	10.54	0.63
Hypoxia Distribution, Case 2	19.78	2.11	15.72	0.72
Lower Body O2, Case 2	25.08	1.71	5.82	0.59

Table 7: CPT discrepancies for the Child network

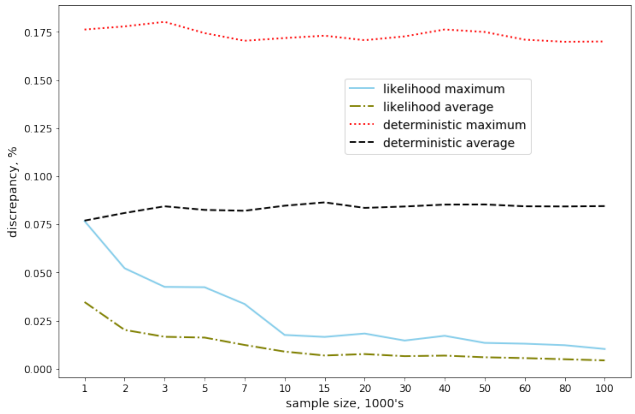


Figure 4: Convergence results for the two-node network.

Summary and Future Work

We have shown how to improve the learning of the parameters of Bayesian networks from uncertain historical data. Our core contributions include an enhancement to the EM algorithm that takes likelihood evidence of historical examples into account, irrespective of the source of the likelihood evidence. We also proved that our algorithm correctly converges to a local maximum of the desired MLE, in spite of the fact that the inference phase is now carried out on a large number of different BNs. Finally, our work also includes an extensive empirical study, showing the importance of explicitly incorporating the uncertainty of historical evidence. This empirical analysis not only demonstrated the large accuracy gap that results when failing to properly account for the uncertainty in the data, but also the possible failure of learning algorithms to converge to the actual parameters when failing to properly account for such uncertainty.

Our planned future work includes validating our approach on a real use case, such as a real diagnosis scenario with historical unstructured data. In such work, we would aim to compare the diagnosis results of a network properly trained on uncertain evidence with the diagnosis results carried out on a network that did not explicitly incorporate such evidence. An additional avenue of future work would be to extend our work to other learning algorithms for BNs, as well as other types of uncertainties. There are several algorithms for learning BN parameters from missing data that use BN-based inference as a part of the learning (an example is the extension to EM proposed by Masegosa, Feelders, and van der Gaag (2016), which can take qualitative expert knowledge into account when learning the parameters). Such algorithms could potentially also be extended in a manner similar to the one we have shown above. Similarly, an algorithm for carrying out inference on other types of uncertain data (such as *fixed probabilistic evidence* uncertainty and *not-fixed probabilistic evidence* uncertainty (Mrad et al. 2015)) could potentially be used to create an EM-based algorithm analogous to the one we have presented here. Of course, any such extensions would have to be analyzed to ensure properties such as correctness and convergence. Finally, as in the EM algorithm, each iteration requires inference on a large number of examples, each on a different network, and as only a local maximum is guaranteed, an important direction would be to either augment our algorithm, or find new algorithms, so as to reduce computation time and provide results closer to the global MLE.

References

Bauer, E.; Koller, D.; and Singer, Y. 1997. Update rules for parameter estimation in Bayesian networks. In *Uncertainty in Artificial Intelligence (UAI)*, 3–13.

Bernardo, J.; and Smith, A. 1994. *Bayesian Theory*. John Wiley and Sons.

Cai, B.; Huang, L.; and Xie, M. 2017. Bayesian Networks in Fault Diagnosis. *IEEE Trans. Industrial Informatics* 13(5).

Dempster, A.; Laird, N.; and Rubin, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B* 39: 1–38.

Elidan, G.; and Friedman, N. 2005. Learning hidden variable

networks: The information bottleneck approach. *Journal of Machine Learning Research* 6: 81–127.

Elidan, G.; Ninio, M.; Friedman, N.; and Shuurmans, D. 2002. Data perturbation for escaping local maxima in learning. In *AAAI/IAAI*, 132–139.

Heckerman, D. 1998. A tutorial on learning with Bayesian networks. In Press, M., ed., *Jordan, M. I. (Ed.), Learning in Graphical Models*, 301–354.

Kahn, C. E.; Roberts, L. M.; Shaffer, K. A.; and Haddawy, P. 1997. Construction of a Bayesian network for mammographic diagnosis of breast cancer. *Computers in Biology and Medicine* 27(1): 19–29.

Koller, D.; and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. MIT Press. ISBN 9780262013192. URL https://books.google.co.in/books?id=7dzpHCHzNQ4C.

Lauritzen, S. 1995. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis* 19: 191–201.

Lauritzen, S.; and Spiegelhalter, D. 1988. Local Computation with Probabilities on Graphical Structures and their Application to Expert Systems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 50(2): 157–224.

Marchetti, S.; and Antonucci, A. 2018. Reliable Uncertain Evidence Modeling in Bayesian Networks by Credal Networks. In *International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 513–518.

Masegosa, A. R.; Feelders, A. J.; and van der Gaag, L. C. 2016. Learning from incomplete data in Bayesian networks with qualitative influences. *International Journal of Approximate Reasoning* 69: 18 – 34. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2015.11.004. URL http://www.sciencedirect.com/science/article/pii/S0888613X15001632.

Mrad, A. B.; Delcroix, V.; Piechowiak, S.; Leicester, P.; and Abid, M. 2015. An Explication of Uncertain Evidence in Bayesian Networks: Likelihood Evidence and Probabilistic Evidence. *Applied Intelligence* 43(4): 802–824. ISSN 0924-669X. doi:10.1007/s10489-015-0678-6. URL https://doi.org/10.1007/s10489-015-0678-6.

Ortiz, L.; and Kaelbling, L. 1999. Accelerating EM: an empirical study. In *Uncertainty in Artificial Intelligence (UAI)*, 512–521.

Pan, R.; Peng, Y.; and Ding, Z. 2006. Belief Update in Bayesian Networks Using Uncertain Evidence. In *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, 441–444.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558604790.

Peng, Y.; Zhang, S.; and Pan, R. 2010. Bayesian network reasoning with uncertain evidence. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 18(5): 539–564.

Song, W.; Yu, J. X.; Cheng, H.; Liu, H.; He, J.; and Du, X. 2012. Bayesian Network Structure Learning from Attribute Uncertain Data. In Gao, H.; Lim, L.; Wang, W.; Li, C.; and Chen, L., eds., *Web-Age Information Management*, 314–321. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-32281-5.

Spiegelhalter, D.; and Best, N. 2003. Bayesian approaches to multiple sources of evidence and uncertainty in complex cost-effectiveness modelling. *Stat Med.* 22(23): 3687–3709.

Spiegelhalter, D.; and Lauritzen, S. 1990. Sequential updating of conditional probabilities on directed acyclic structures. *Networks* 20: 579–605.

Spiegelhalter, D. J.; and Cowell, R. G. 1992. Learning in probabilistic expert systems. In *J. M. Bernardo and J. 0. Berger and A. P. Dawid and A. F. M. Smith (Eds.), Bayesian Statistics 4*, 447–466.

Thiesson, B.; Meek, C.; and Heckerman, D. 2001. Accelerating EM for large databases. *Machine Learning* 45(3): 279–299.

Valtorta, M.; Kim, Y.-G.; and Vomlel, J. 2002. Soft evidential update for probabilistic multi-agent systems. *International Journal or Approximate Reasoning* 29(1): 71–106.

Wasserkrug, S.; Krüger, M.; Feldman, Y. A.; Shindin, E.; and Zeltyn, S. 2019. What's Wrong with My Dishwasher: Advanced Analytics Improve the Diagnostic Process for Miele Technicians. *INFORMS J. Applied Analytics* 49(8).